386 | DEBUG
Reference Card

Phar Lap Software, Inc.
60 Aberdeen Ave., Cambridge, MA 02138
(617) 661-1510
FAX (617) 876-2972
dox@pharlap.com
tech-support@pharlap.com

Conventions

`courier`	indicates command line syntax
italics	indicate a name or value that must be entered by the user
[ ]	(brackets) indicate optional items or information
|	(vertical bar) separates alternative items

Commands-Functional Listing

386|DEBUG Options

Redirect Debugger I/O

To serial channel #1	`COM1`
To serial channel #2	`COM2`
To console (CON)	`CON`

Memory Models

Show current model	`M?`
Assume small model	`MS`
Assume compact model	`MC`
Assume medium model	`MM`
Assume large model	`ML`

Address Formation

Assume protected mode addresses	`PROT`
Assume real mode addresses	`REAL`

continued ↗

Breakpoints and Watchpoints

Set

Set breakpoints	BP address
Set watchpoints	WP addr datalen [r|w]

Clear

Clear breakpoints	BC breakpoints | *
Clear watchpoints	WC watchpoints | *

List

List breakpoints	BL
List watchpoints	WL

Enable

Enable breakpoints	BE breakpoints | *
Enable watchpoints	WE watchpoints | *

Disable

Disable breakpoints	BD breakpoints | *
Disable watchpoints	WD watchpoints | *

Display Active Procedure Calls

Display call stack	K [arg_count]
Display call stack and arguments	KA

Display System Tables

Dump Descriptor Tables

Global Descriptor Table	DG [range]
Interrupt Descriptor Table	DI [range]
Local Descriptor Table	DL [range]

Page Table Info

Display at address	PI address
Display at linear address	PL linaddr

Dump Task State Segments

Display TSS at selector	DTSS [selector]
Display 386 TSS at address	DTSS32 address
Display 286 TSS at address	DTSS16 address

Examine/Change Memory

As Data

Hex bytes	DB [range]
Hex words	DW [range]
Hex double	DD [range]
ASCII	DA [range]
Last format	D [range]
As Floating Point (8-byte)	DQ [range]
(4-byte)	DS [range]
(10-byte)	DT [range]

As Code (Unassembled)

Unassemble memory	U [range]
In 16-bit mode	U16 [range]
In 32-bit mode	U32 [range]

continued ↗

Examine/Change Memory (cont.)

Enter (Change) Memory

By bytes	E *address value(s)*
By bytes	EB *address value(s)*
By words	EW *address value(s)*
By doublewords	ED *address value(s)*

Memory Block Operations

Compare blocks	C *range address*
Move block	M *range address*
Fill block	F *range value(s)*
Search block for string	S *range value(s)*

Examine/Change Registers

Display or change registers	R [*register*] [*value*]
Display extended registers	RX
Display 80387 registers	R87
Display Weitek registers	R67
Display Weitek registers	R67D
Display Weitek registers in single precision	R67S

Execute and Trace

Go (and set breakpoints)	G [=*address*] [*address(es)*]

Trace (skipping CALL instructions)

Verbose	P [=*address*] [*count*]
Quiet	PQ [=*address*] [*count*]

Trace (single-step)

Verbose	T [=*address*] [*count*]
Quiet	TQ [=*address*] [*count*]

I/O

Port

Input from port	I *port*
Output to port	O *port value*

Interactive Commands

Display debugger commands	? [char]
Hex Arithmetic	H *value value*
Quit	Q

Symbol Table

Display symbol table	X [*symbol*]
Relocate symbol table	XR *old_sel new_sel*
Display segment names	XS
Locate symbol at address	XW *address*

Printed on recycled paper.

Commands-Alphabetical Listing

<table>
<tr><td>? [char]</td><td>Display debugger commands</td></tr>
<tr><td>BC breakpoints | *</td><td>Clear breakpoints</td></tr>
<tr><td>BD breakpoints | *</td><td>Disable breakpoints</td></tr>
<tr><td>BE breakpoints | *</td><td>Enable breakpoints</td></tr>
<tr><td>BL</td><td>List all breakpoints</td></tr>
<tr><td>BP address</td><td>Set breakpoints</td></tr>
<tr><td>C range address</td><td>Compare memory</td></tr>
<tr><td>COM1</td><td>Redirect I/O to COM1</td></tr>
<tr><td>COM2</td><td>Redirect I/O to COM2</td></tr>
<tr><td>CON</td><td>Return I/O to console</td></tr>
<tr><td>D [range]</td><td>Dump memory (repeating last format)</td></tr>
<tr><td>DA [range]</td><td>Dump memory in ASCII</td></tr>
<tr><td>DB [range]</td><td>Dump memory as individual bytes</td></tr>
<tr><td>DD [range]</td><td>Dump memory as hex doublewords</td></tr>
<tr><td>DF [range]</td><td>Dump memory as floating point (4-byte)</td></tr>
<tr><td>DG [range]</td><td>Dump GDT</td></tr>
<tr><td>DI [range]</td><td>Dump IDT</td></tr>
<tr><td>DL [range]</td><td>Dump LDT</td></tr>
<tr><td>DQ [range]</td><td>Dump memory as floating point (8-byte)</td></tr>
<tr><td>DS [range]</td><td>Dump memory as floating point (4-byte)</td></tr>
<tr><td>DT [range]</td><td>Dump memory as floating point (10-byte)</td></tr>
<tr><td>DTSS [selector]</td><td>Dump task state segment</td></tr>
<tr><td>DTSS16 address</td><td>Dump 286 TSS at address</td></tr>
<tr><td>DTSS32 address</td><td>Dump 386 TSS at address</td></tr>
<tr><td>DW [range]</td><td>Dump memory as hex words</td></tr>
<tr><td>E address value(s)</td><td>Enter (change) memory by bytes</td></tr>
<tr><td>EB address value(s)</td><td>Enter (change) memory by bytes</td></tr>
<tr><td>ED address value(s)</td><td>Enter (change) memory by doublewords</td></tr>
<tr><td>EW address value(s)</td><td>Enter (change) memory by words</td></tr>
<tr><td>F range value(s)</td><td>Fill memory</td></tr>
<tr><td>G [=address] [address(es)]</td><td>Go (and set temporary breakpoints)</td></tr>
<tr><td>H value value</td><td>Hex arithmetic</td></tr>
<tr><td>I port</td><td>Read byte value from input port</td></tr>
<tr><td>K [arg_count]</td><td>Display procedure call stack</td></tr>
<tr><td>KA</td><td>Display call stack with arguments</td></tr>
<tr><td>M range address</td><td>Move memory</td></tr>
<tr><td>M?</td><td>Show current memory model</td></tr>
<tr><td>MC</td><td>Select compact memory model</td></tr>
<tr><td>ML</td><td>Select large memory model</td></tr>
<tr><td>MM</td><td>Select medium memory model</td></tr>
<tr><td>MS</td><td>Select small memory model</td></tr>
<tr><td>O port value</td><td>Output data byte to port</td></tr>
<tr><td>P [=address] [count]</td><td>Trace across CALL instructions</td></tr>
<tr><td>PI address</td><td>Display page table info</td></tr>
<tr><td>PL linaddr</td><td>Display page table info</td></tr>
<tr><td>PQ</td><td>Trace quietly across CALL instructions</td></tr>
<tr><td>PROT</td><td>Assume protected mode addresses</td></tr>
<tr><td>Q</td><td>Quit</td></tr>
<tr><td>R [register [value]]</td><td>Display or change registers</td></tr>
<tr><td>R67</td><td>Display Weitek registers in double precision</td></tr>
<tr><td>R67D</td><td>Display Weitek registers in double precision</td></tr>
<tr><td>R67S</td><td>Display Weitek registers in single precision</td></tr>
<tr><td>R87</td><td>Display 80387 registers</td></tr>
<tr><td>REAL</td><td>Assume real mode addresses</td></tr>
<tr><td>RX</td><td>Display extended registers</td></tr>
<tr><td>S range value(s)</td><td>Search memory for string</td></tr>
<tr><td>T [=address] [count]</td><td>Trace (single-step)</td></tr>
<tr><td>TQ [=address] [count]</td><td>Trace quietly</td></tr>
<tr><td>U [range]</td><td>Unassemble memory</td></tr>
<tr><td>U16 [range]</td><td>Unassemble memory in USE16 segment</td></tr>
<tr><td>U32 [range]</td><td>Unassemble memory in USE32 segment</td></tr>
<tr><td>WC watchpoints | *</td><td>Clear watchpoints</td></tr>
<tr><td>WD watchpoints | *</td><td>Disable watchpoints</td></tr>
<tr><td>WE watchpoints | *</td><td>Enable watchpoints</td></tr>
<tr><td>WL</td><td>List all watchpoints</td></tr>
<tr><td>WP addr datalen [r |w]</td><td>Set watchpoints</td></tr>
<tr><td>X [symbol]</td><td>Display symbol table</td></tr>
<tr><td>XR old_sel new_sel</td><td>Relocate symbols to another selector</td></tr>
<tr><td>XS</td><td>Display segment names</td></tr>
<tr><td>XW address</td><td>Locate symbol at address</td></tr>
</table>